University of Toronto

ECE 1504

Project

Interpretability and Performance Analysis of Deep Neural Structures on Time Series Classification Task

Prepared by:

Berk Gorgulu
1005079898

Shiva Saxena
1005609370

December 17, 2018

1 Introduction

Time series data mining is an important task with many challenging applications including finance, science, medicine and multimedia. Classification is the primary goal in many of these applications. Although, deep neural networks are very successful in image classification, there has been very few work related to time-series classification. The main reason behind this is that the deep neural networks are considered as black box models which means they are not interpretable. Therefore, for classification task, it is not easy to determine why an instance is assigned to a certain class when the deep learning models are used.

In this project, we aim to apply deep learning methods on time-series classification task by comparing various network structures [5] in terms of their classification performances and interpretability. Interpretability of the networks is examined by visual inspection of the layer activations and by implementing advanced interpretation methods such as **Sensitivity Analysis**, **Simple Taylor Expansion** and **Layer-wise Relevance Propagation** suggested in [4].

2 Classification and Supervised Methods

In this section, we implement MLP, CNN, ResNet, DenseNet structures for time series classification task. We train each model on 9 time series datasets namely Beef, CBF, Coffee, GunPoint, ShapesAll, Strawberry, TwoPatterns, Wafer and Yoga datasets from UCR database [1]. We then evaluate and compare test performance of the models on each dataset respectively.

We start with multi-layer perceptrons (MLP) which can be considered as the base model in this analysis. We costructed four different multi-layer perceptron with varying number of hidden layers and hidden units including the model suggested by [5]. You can find the model specifications and complete experimental results in Table 3 and Table 2 respectively in Appendix A. From these expermients, we see that the model suggested by [5] provides the best performance among four alternatives.

We advance our analysis by implementing deep convolutional model (CNN) which are more convenient models for time-series classification. Filters of the convolutions can be thought of as small subsequences that distinguish each class of time-series from the others. After experimenting with many different convolutional networks, three models are selected as the candidate CNN models including the model suggested by [5]. In order to prevent over-fitting and speed-up the convergence batch-normalization is used after each convolutional layer. Global average pooling is used at the end of the last convolutional layer instead of pooling operation after each convolutional layer as suggested by [5]. Table 3 and Table 5 in Appendix A provides the parameter settings and classification performances of each candidate methods. From the analysis, we see that the third model clearly performs the best. Therefore, this model is used in the following sections and referred as the deep convolutional network.

Residual Networks(ResNets) are introduced by [2] allowing to build deeper models without experiencing vanishing gradient problem. In order to demonstrate the performance of residual learning on the time-series classification, we implemented a deep learning model with residual structure suggested in [5]. Network consists of three residual block and a global average pooling. Each residual block contains three convolutional layer with filter sizes 8,5,3 and 128,256,128 number of filters respectively. Lastly, we implemented densly connected network structure (DenseNets) suggested by [3] to examine its performance in the time series domain. In our model, each densely connected block contains three convolutional layer with filter sizes 8,5,3 and 128, 256, 128 number of filters respectively. We implemented batch normalization for faster convergence and dropout for regularization in both models.

Table 1 provides the classification performances of the best MLP, CNN, ResNet and DenseNet models selected above. As expected, CNN, ResNet and DenseNet perfom better than the MLP which shows the similarity between the image classification and time series classification tasks. In addition, CNN performs very strongly and even though it has less parameters than ResNet and DenseNet, it provides a comparable performance.

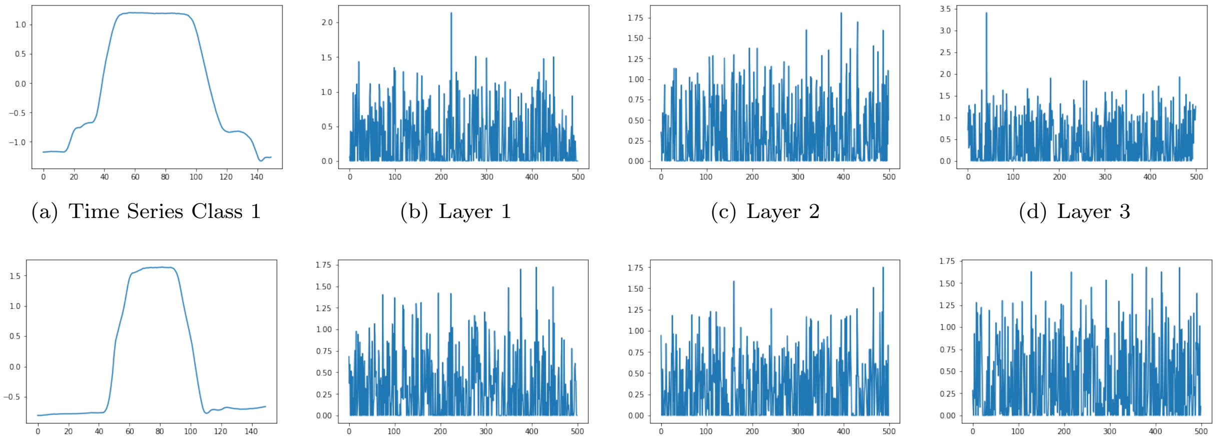
Table 1: Comparison of the classification accuracies of the MLP, CNN, ResNet and DenseNet on 9 datasets from UCR database

	MLP	CNN	ResNet	DenseNet
Beef	**0.867**	0.8	0.833	0.767
CBF	0.86	**1**	**1**	0.99
Coffee	**1**	**1**	**1**	**1**
Gun Point	0.953	**1**	0.993	**1**
ShapesAll	0.753	0.835	**0.885**	0.877
Strawberry	0.96	0.965	**0.970**	0.964
Two Patterns	0.888	0.897	0.948	**0.956**
Wafer	0.995	**0.996**	0.993	0.995
Yoga	0.852	0.865	**0.874**	**0.874**

3 Interpretation via Visualization

In the previous section, classification performances of different deep network structures are compared. In this section interpretabiliy of these methods are evaluated by visualizing both, the layer activations and the network parameters (weights). We select Gun Point dataset to be used in interpretability analysis since it is possible for us to distinguish two classes of time series visually. Class 1 time series is classified differently from class 2 time series due to the small sub-peaks before reaching the global peak and small stop before settling in the lower level. A representative example of each class of time series is provided in Figure 1(a) and 1(e).

The analysis again starts with the most basic network structure. Layer activations of MLP for two different class of time series are provided in Figure 1. None of the layer activations provide any insight about the classification procedure and how the network distinguishes one class of time series from another. This is an expected outcome, because MLP is not specialized for any specific input type and it has a highly complex structure. In addition to the layer activations, visualization of the weights can be found in Figure 11 in Appendix B.



(a) Time Series Class 1

(b) Layer 1

(c) Layer 2

(d) Layer 3

(e) Time Series Class 2

(f) Layer 1

(g) Layer 2

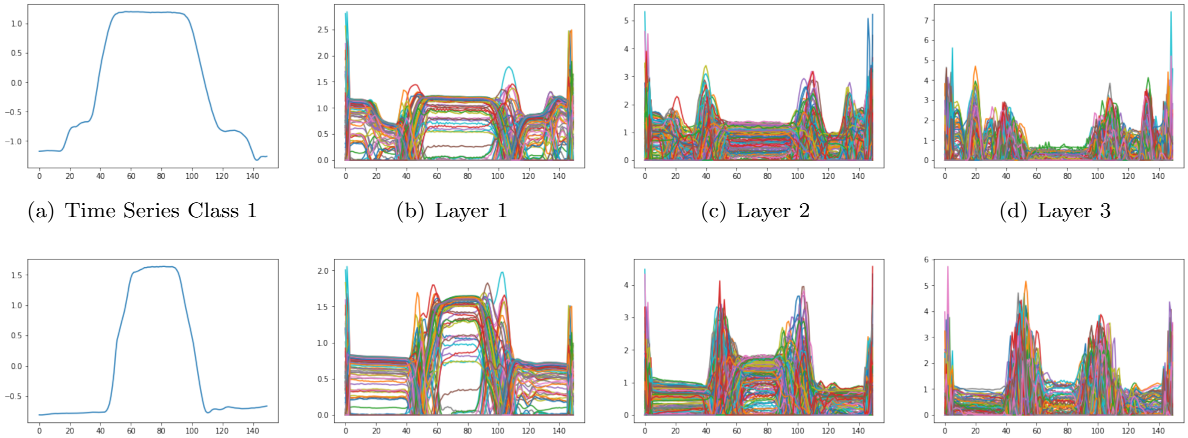
(h) Layer 3

Figure 1: Visualization of MLP Layer Activations

Deep convolutional networks are mainly developed for image classification tasks. They are capable of capturing the relationships between neighbor pixels. Therefore, they can distinguish important patterns and can handle feature shifts in the images. As the time series can be thought of as a 1-D image, we expect deep convolutional networks to identify the important patterns in time series as well.

Layer activations of CNN are provided in Figure 2. Due to the weight sharing in deep convolutional networks, layer activations provide interpretable outputs. Class 1 time series has additional sub-peaks in both left and right side of the main peak. It can be seen from the layer activations that network understands this difference and tries to make a classification decision based on those parts of the time-series. Moreover, network tries to absorb these sub peaks in class 1 and exaggerate the same locations in class 2.

In addition to the layer activations, kernels of the convolutional layers are expected to learn smaller subsequences called "shapelets" that maximize the distance between class 1 and class 2 time series. Visualization of these kernels can be found in Figure 12 in Appendix B.



(a) Time Series Class 1 (b) Layer 1 (c) Layer 2 (d) Layer 3

(e) Time Series Class 2 (f) Layer 1 (g) Layer 2 (h) Layer 3

Figure 2: Visualization of Deep Convolutional Network Layer Activations

ResNets and DenseNets utilizes convolutional neural networks, therefore we expect their activations to be somewhat similar to the convolutional neural networks. However, for the sake of complete analysis, visualization of the layer activations of ResNets and DenseNets are also provided in the Appendix B.

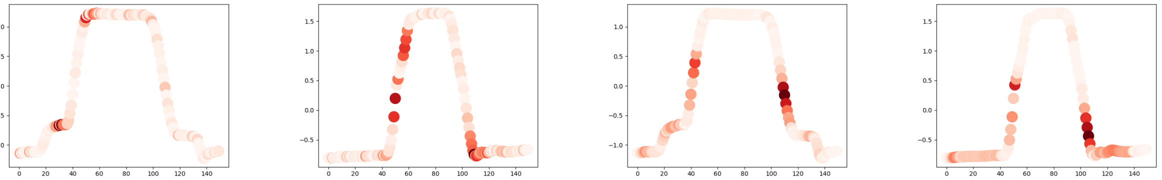
4 Advanced Interpretation Techniques

Compared to the multi-layer perceptrons, deep convolutional networks are specialized network structures and their layer activations are more interpretable. However, the visual inspection does not always provide easily interpretable results even for deep convolutional networks. As network gets deeper and deeper, the concepts learned by the network become highly non-linear and abstract, which consequently makes it harder and sometimes even impossible to interpret the networks by visual inspection. Therefore, some advanced interpretation techniques are suggested for understanding the deep neural networks. In this section, we focus on the Sensitivity Analysis, Simple Taylor Decomposition and Layer-wise Relevance Propagation methods introduced by [4]. We implement these methods for interpreting the MLP and CNN structures trained on the Gun Point dataset. Moreover, the results of the analysis are provided in the form of heat maps. In these heat maps, important parts of the input space are represented with the colors red and blue where red represents the importance to the first class and blue represents the importance to the second class.

4.1 Sensitivity Analysis

Sensitivity analysis is introduced as the simplest method for studying the interpretability. It is based on the models locally evaluated gradients which comes from the assumption, "important features are affected more with the changes in the output". In this analysis $R_i(x) = \left(\frac{\delta f}{\delta x_i}\right)^2$ is used as a sensitivity measure as suggested in [4]. Therefore, sensitivity analysis solely checks whether a point is important or not without considering which class it is important for. As a result of this fact, heat maps that we generate for sensitivity analysis only contain red colored points.

To perform comparative study, sensitivity analysis is applied to two main network structures. Figure 3 provides the sensitivity analysis results for MLP and CNN on the Gun Point dataset. From the figures it can be deduced that both MLP and CNN capture some of the important points from the input time series. However, these points are very sparse and therefore they do not provide a meaningful result as the relevance is calculated with a huge assumption.

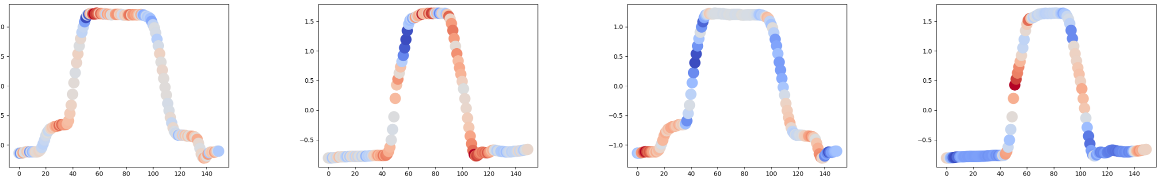


(a) MLP Class 1 (b) MLP Class 2 (c) CNN Class 1 (d) CNN Class 2

Figure 3: Sensitivity Analysis of MLP and CNN on Gun Point Dataset

4.2 Simple Taylor Decomposition

Simple taylor decomposition is a method that explains the models decision by decomposing the output value as a sum of relevance scores. As the name of the method suggest first order approximation is used in the calculation of the relevance scores. In this method relevance is computed as $R_i(x) = x_i \frac{\delta f}{\delta x_i}$. Figure 4 provides the heat maps constructed by Simple Taylor Decomposition for CNN and MLP structures. In addition to the magnitude of the derivative, simple taylor decomposition also takes sign and magnitude of the original input into account. The heat maps are somewhat similar to the heat maps constructed by Sensitivity Analysis but they are more accurate and descriptive. Therefore, this method provides better results than the sensitivity analysis since the relevance not only depends on the change in the input but also depends on the input itself.



(a) MLP Class 1 (b) MLP Class 2 (c) CNN Class 1 (d) CNN Class 2

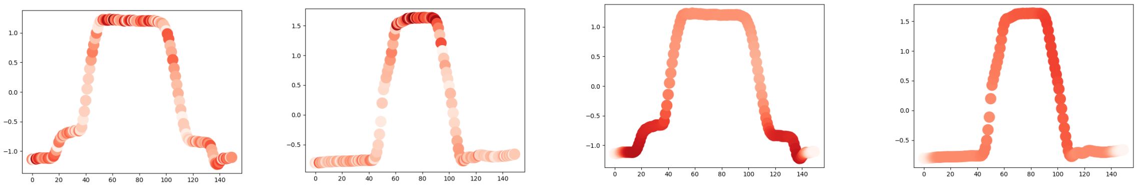
Figure 4: Simple Taylor Decomposition of MLP and CNN on Gun Point Dataset

4.3 LRP (Layerwise Relevance Propagation)

LRP is an interpretation method that uses the feed-forward graph structure. Starting from the output, network distributes the relevance to the previous layers in a reverse direction. Relevance score is conserved in each layer and distributed to previous layer according to its activation and weights. Figure 5 provides the LRP results of the MLP and CNN on the same time series from previous sections. It can be seen that unlike sensitivity analysis, the important points are highly concentrated and they form interpretable patterns.

CNN provides 100% train and test accuracy on the Gun Point dataset, therefore we expect CNN to identify the important parts of the inputs. Figure 5 provides a heat map created by using the standard version of the LRP where $\alpha = 1$. It can be seen that deep convolutional networks understands the distinguishing patterns between both classes of time series perfectly. From our prior knowledge, we know that sub-peaks are important for class 1 time series and LRP demonstrates that perfectly by highlighting the sub-peaks with the darkest shade of red. On the other hand, for class 2 we do not see any particular emphasized regions.

α parameter determines the effect of negative weights on the relevance propagation. Depending on the application, different α values provide better interpretability. Figure 6 provides heat maps of the inputs with the changing *alpha* values. Note that $\alpha = 1$ only considers the relevance to class 1 and as we increase α relevant points for class 2 starts to appear in the heat maps.



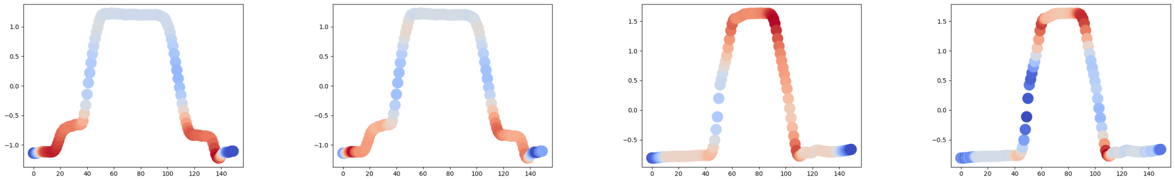
(a) MLP Class 1

(b) MLP Class 2

(c) CNN Class 1

(d) CNN Class 2

Figure 5: LRP of MLP and CNN on Gun Point Dataset with Importance $\alpha = 1$



(a) Class 1, $\alpha = 10$

(b) Class 1, $\alpha = 20$

(c) Class 2, $\alpha = 10$

(d) Class 2, $\alpha = 20$

Figure 6: LRP of CNN with varying α

5 Conclusion & Future Work

In the first part of the project we analyzed the performance of different deep neural structures on the time series classification task. As a result of our analysis we observed that CNN's are more suitable network structures for time series classification task in terms of classification performance. Moreover, the performance can be improved by using residual or dense structures. In the second part variety of supervised network activations are inspected visually to understand the concepts learned by each model. Lastly, in the third part some advanced interpretation methods introduced by [4] are implemented and applied to MLP and CNN structures. From our study, we conclude that convolutional networks are the most interpretable models and LRP performs as the best interpretation tool.

While working on time series classification, we realized that the sequence and the location of the patterns carry vast importance. Therefore, as a further work we would like to implement a deep convolutional network followed by some recurrent structure such as LSTM's or GRU's to not only identify the important patterns but also learn their relative ordering.

References

[1] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, 31:606–660, 2017.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[3] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, volume 1, page 3, 2017.

[4] G. Montavon, W. Samek, and K.-R. Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 2017.

[5] Z. Wang, W. Yan, and T. Oates. Time series classification from scratch with deep neural networks: A strong baseline. In *Neural Networks (IJCNN), 2017 International Joint Conference on*, pages 1578–1585. IEEE, 2017.

Appendices

A Tables

Table 2: Comparison of the classification performances of candidate multi-layer perceptrons

	MLPv1	MLPv2	MLPv3	MLPv4
Beef	**0.867**	0.8	0.8	**0.867**
CBF	0.849	0.86	**0.911**	0.86
Coffee	**1**	**1**	**1**	**1**
Gun Point	0.92	0.927	0.933	**0.953**
ShapesAll	0.73	0.748	0.74	**0.753**
Strawberry	0.949	0.949	0.959	**0.96**
Two Patterns	0.863	0.88	0.879	**0.888**
Wafer	0.992	0.994	0.993	**0.995**
Yoga	0.832	0.841	0.841	**0.852**

Table 3: Multi-layer perceptron candidate model properties

	Number of Hidden Layers	Number of Hidden Units	Learning Rate	Activation Function	Dropout	Optimizer
MLPv1	1	125	0.01	Relu	0.8	Adam
MLPv2	2	100-100	0.01	Relu	0.8	Adam
MLPv3	3	125-62-31	0.01	Relu	0.8	Adam
MLPv4	3	500-500-500	0.01	Relu	0.8	Adam

Table 4: CNN candidate model properties

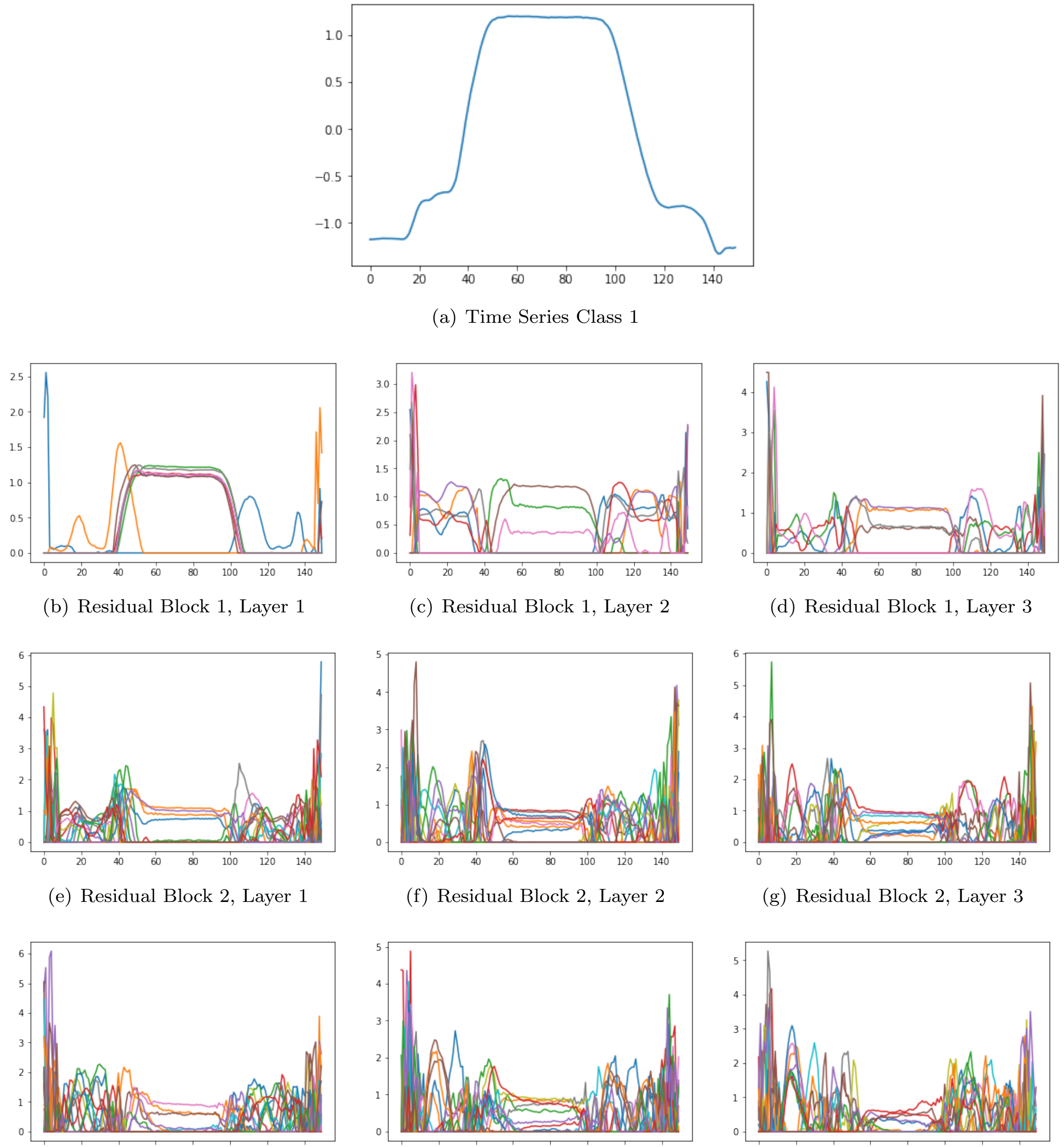
	Number of Hidden Layers	Number of Filters	Filter Size	Learning Rate	Activation Function	Dropout	Batch Normalization	Optimizer
CNNv1	1	200	7	0.001	Relu	0.8	Yes	Adam
CNNv2	4	75-50-25-10	10-20-20-10	0.001	Relu	0.8	Yes	Adam
CNNv3	3	128-256-128	8-5-3	0.001	Relu	0.8	Yes	Adam

Table 5: Comparison of the classification performances of candidate deep convolutional networks

	CNNv1	CNNv2	CNNv3
Beef	**0.9**	0.833	0.8
CBF	0.77	0.911	**1**
Coffee	**1**	**1**	**1**
Gun Point	0.94	0.98	**1**
ShapesAll	0.692	0.698	**0.835**
Strawberry	0.96	0.961	**0.965**
Two Patterns	0.852	**0.988**	0.897
Wafer	0.992	0.995	**0.996**
Yoga	0.735	0.818	**0.865**

B Additional Activation and Weight Visualization

B.1 Activations



(a) Time Series Class 1

(b) Residual Block 1, Layer 1

(c) Residual Block 1, Layer 2

(d) Residual Block 1, Layer 3

(e) Residual Block 2, Layer 1

(f) Residual Block 2, Layer 2

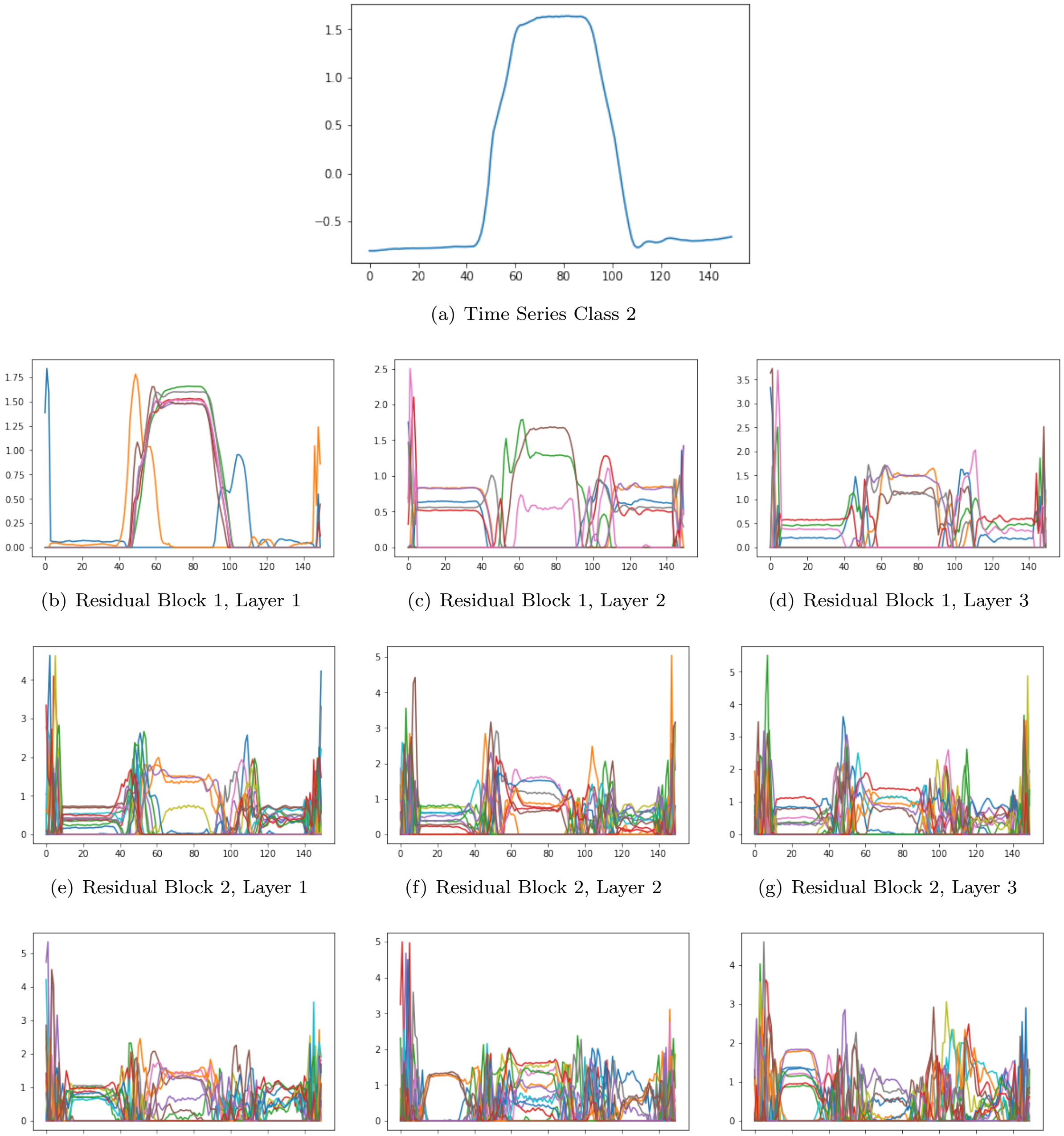
(g) Residual Block 2, Layer 3

(h) Residual Block 3, Layer 1

(i) Residual Block 3, Layer 2

(j) Residual Block 3, Layer 3

Figure 7: Visualization of residual network layer activations of class 1 time series



(a) Time Series Class 2

(b) Residual Block 1, Layer 1

(c) Residual Block 1, Layer 2

(d) Residual Block 1, Layer 3

(e) Residual Block 2, Layer 1

(f) Residual Block 2, Layer 2

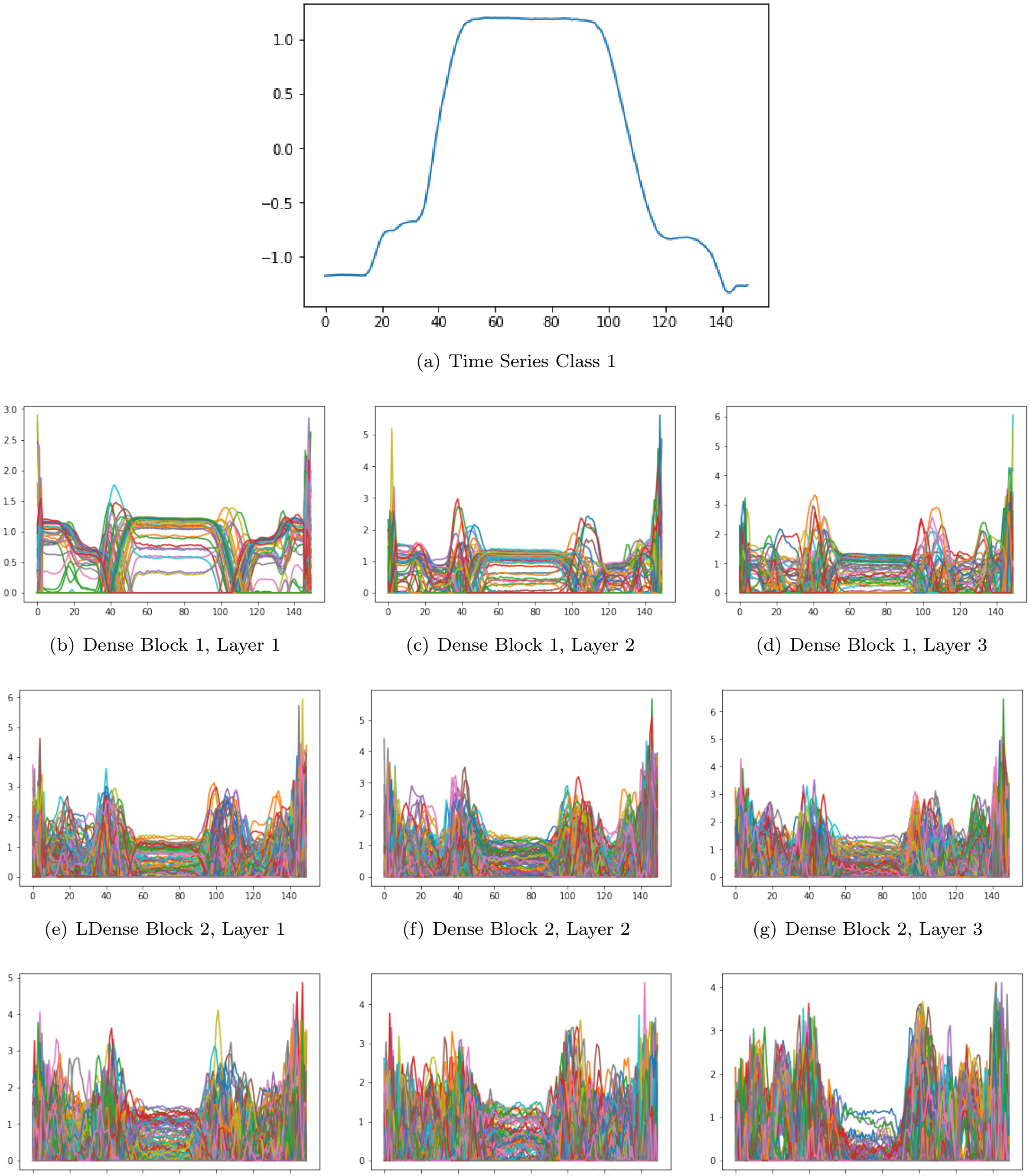
(g) Residual Block 2, Layer 3

(h) Residual Block 3, Layer 1

(i) Residual Block 3, Layer 2

(j) Residual Block 3, Layer 3

Figure 8: Visualization of residual network layer activations of class 2 time series



(a) Time Series Class 1

(b) Dense Block 1, Layer 1

(c) Dense Block 1, Layer 2

(d) Dense Block 1, Layer 3

(e) LDense Block 2, Layer 1

(f) Dense Block 2, Layer 2

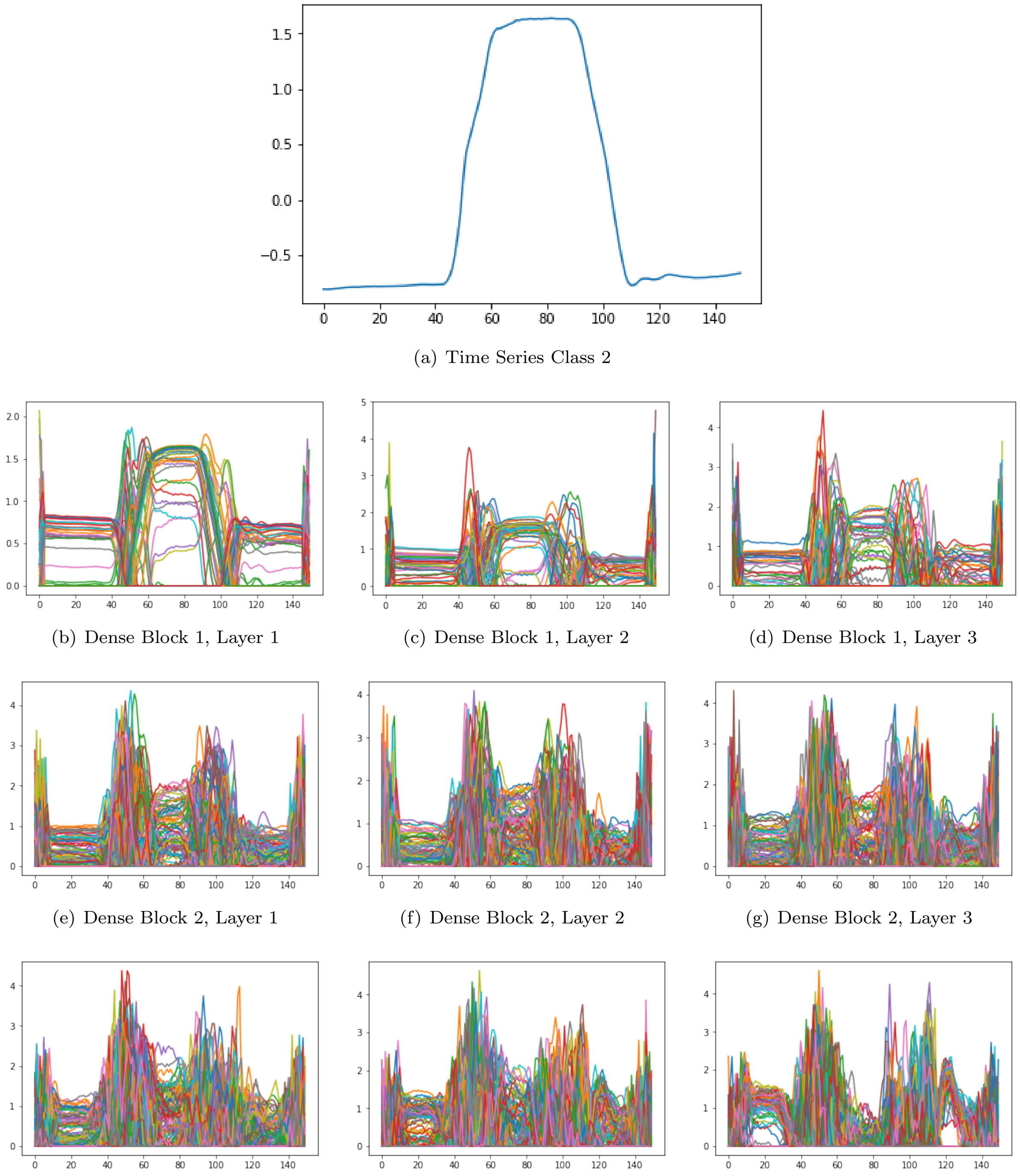
(g) Dense Block 2, Layer 3

(h) Dense Block 3, Layer 1

(i) Dense Block 3, Layer 2

(j) Dense Block 3, Layer 3

Figure 9: Visualization of DenseNet Layer Activations of Class 1 Time Series



(a) Time Series Class 2

(b) Dense Block 1, Layer 1

(c) Dense Block 1, Layer 2

(d) Dense Block 1, Layer 3

(e) Dense Block 2, Layer 1

(f) Dense Block 2, Layer 2

(g) Dense Block 2, Layer 3

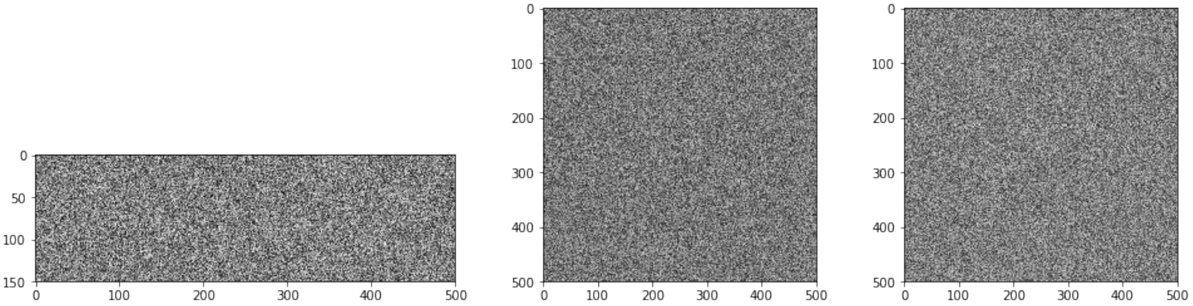
(h) Dense Block 3, Layer 1

(i) Dense Block 3, Layer 2

(j) Dense Block 3, Layer 3

Figure 10: Visualization of DenseNet Layer Activations of Class 2 Time Series

B.2 Weights

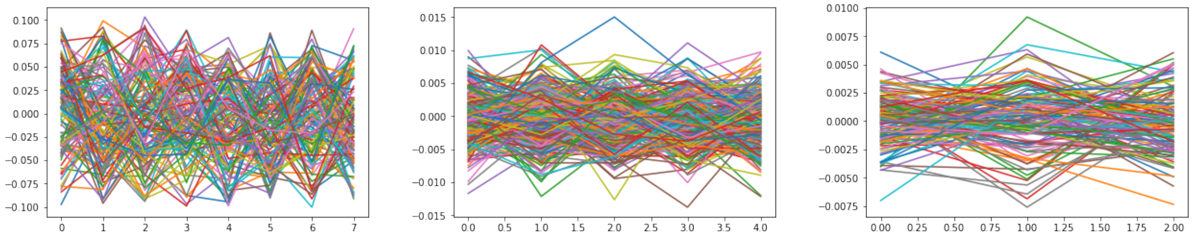


(a) Layer 1

(b) Layer 2

(c) Layer 3

Figure 11: Visualization of MLP Weights



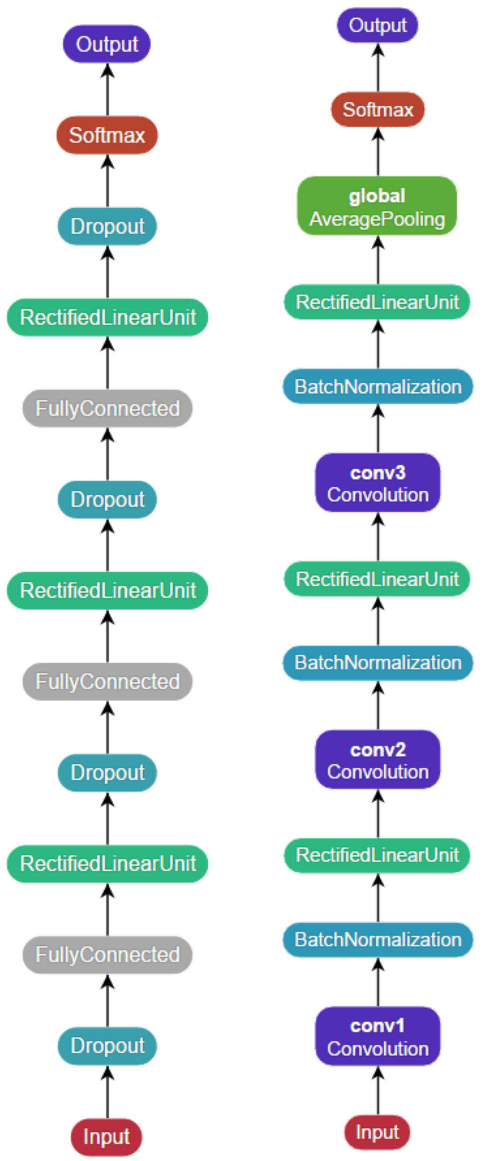
(a) Layer 1

(b) Layer 2

(c) Layer 3

Figure 12: Visualization of Deep Convolutional Network Weights

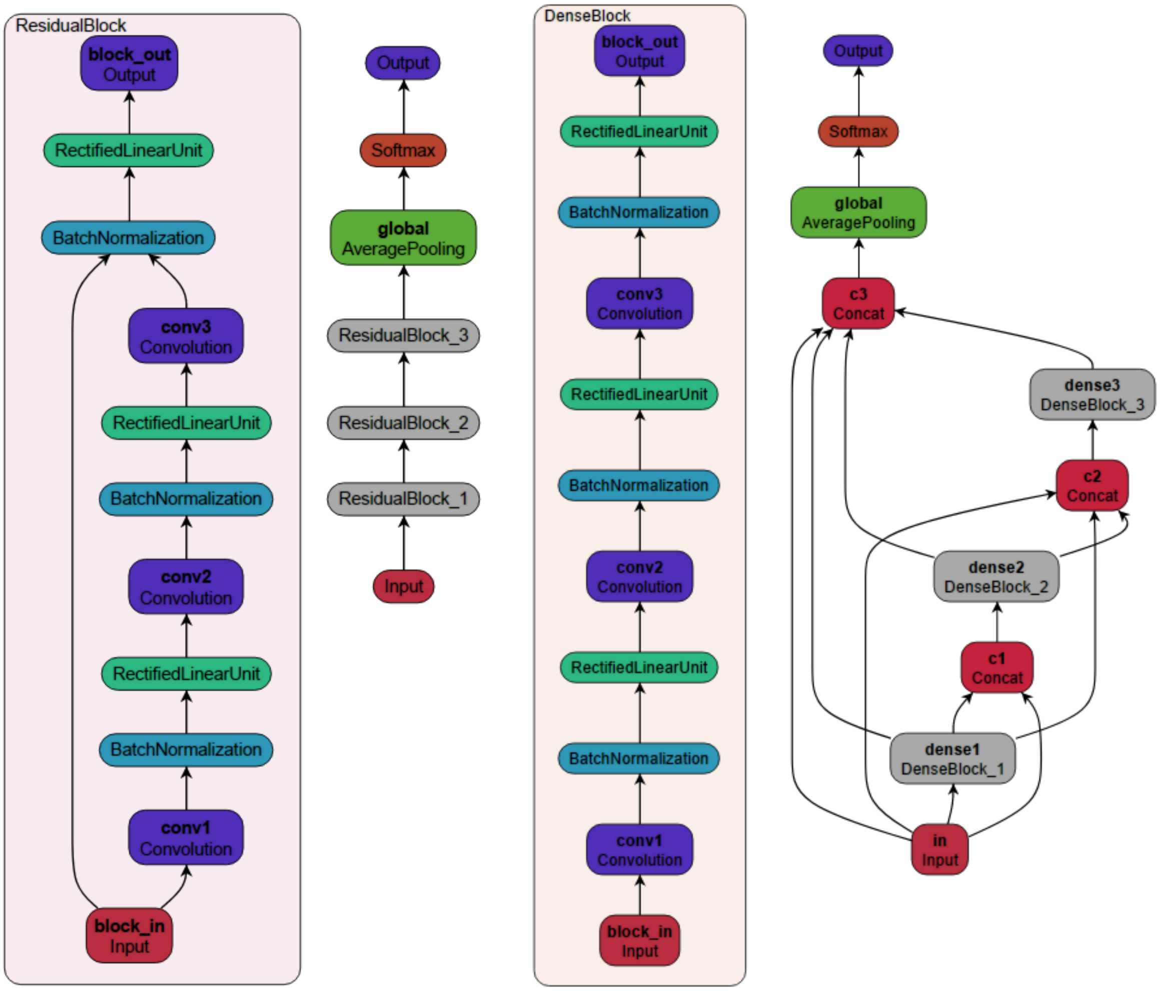
C Model Figures



(a) MLP

(b) Deep CNN

Figure 13: Visual Models of MLP and CNN



(a) Residual Network

(b) Densely Connected Network

Figure 14: Residual network and densely connected network